

1 Integration Approaches

Scalable Data Engineering

Why data integration?

Data Growth

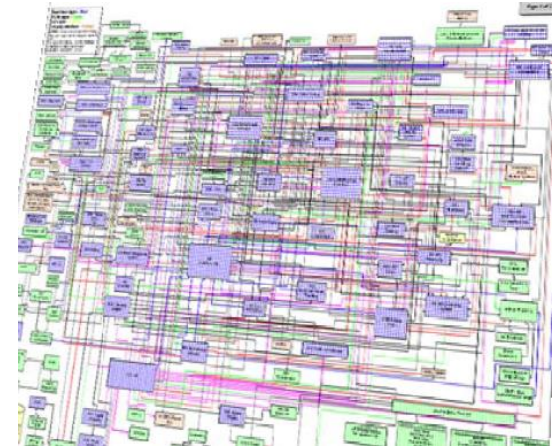
- The Global StorageSphere installed base of storage capacity reached 6.7ZB in 2020
- Estimated 163 ZB in 2025
- 64.2ZB of data was created or replicated [[IDC](#)]

Distributed Information

- Multiple heterogeneous systems und applications
- Technical and organizational reasons

Necessity of Integration

- “Application infrastructure and middleware market” growth by **7.3%** from **33.82 Mrd. USD** in **2020** to **51.28 Mrd USD** in **2025** [<https://www.mordorintelligence.com>]
- Integration is one of the biggest/costly challenges for IT companies (**40%** of budgets spend for integration software and projects)



[Albert Maier: Master Data Management, Gastvorlesung, TU Dresden, AVIS, WS08/09.]

Reasons for Distribution

- Database technology was created as a way for integration and should have solved the problem
→ *single company-database*
- **Problems:** forward-looking design, requirements cannot be fulfilled completely (restructuring, advancement, pragmatism)
→ *department-/specialty-databases*

State of Distributed Information Systems

- Company data is stored in several databases
- Distribution not disjoint, mostly heterogeneous representations
- Complex workflows: change of data in application A → change of data in applications B,C,....

→ Assure consistency ←

Why analytics?

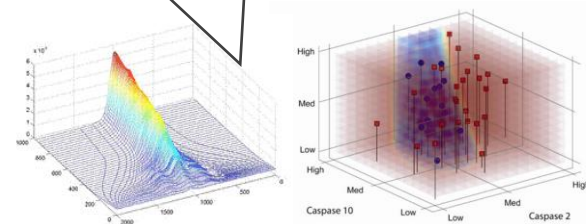
Analytic Scenarios

How many female WWF fans under the age of 30 visited the Toyota community over the last 4 days and saw a Class A ad?

31	29	3
22	42	26
19	14	11

**Traditional Data Analytics/
Reporting**
(aggregation queries)

How are these people similar to those that visited Nissan?



Advanced Data Analytics
(Clustering, Classification, Forecasting,
Association Rules)

Homogeneity

- Homogeneity (greek homo/homoios = uniform/similar) describes the uniformity of things
- In computer science: uniformity of characteristics, technologies, systems, or concepts

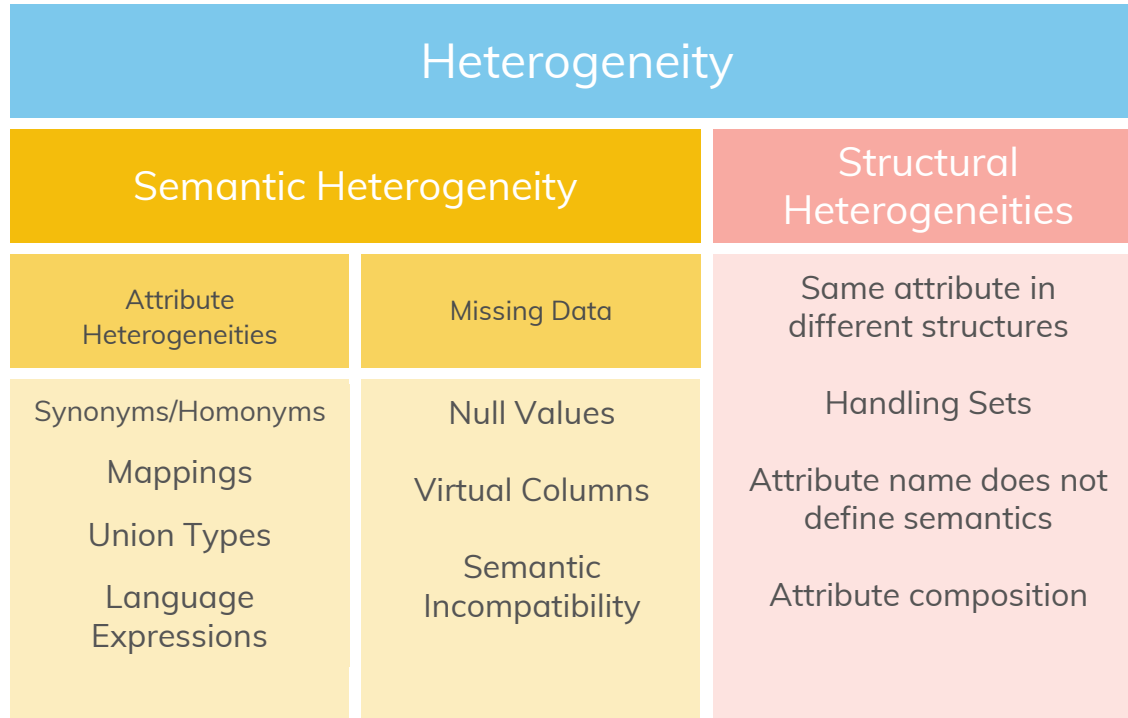
Heterogeneity

- Heterogeneity (also inhomogeneity) describes the difference of things

Integration

- Integration (lat. Integer = whole) describes the creation of a whole.
- In computer science: the global consolidation of certain local integration objects (systems, applications, data sets, or functions) using a defined type of integration

Types of Heterogeneity



Synonyms/Homonyms

- Different attribute names with same semantics (earnings $\leftrightarrow$ profit)

Mappings

- Related attributes of different schemata differ in transformation/derivation of data
- For example: temperature in °C and K

Union Types

- Attributes of different schemata use different datatypes to represent the same information (1024 $\leftrightarrow$ "1024")

Language

- Names or values of identical attributes are expressed in different languages (blau $\leftrightarrow$ blue)

Null-Values (Missing Values)

- The value of that attribute does not exist. This is especially problematic when the attribute does not allow null-values.

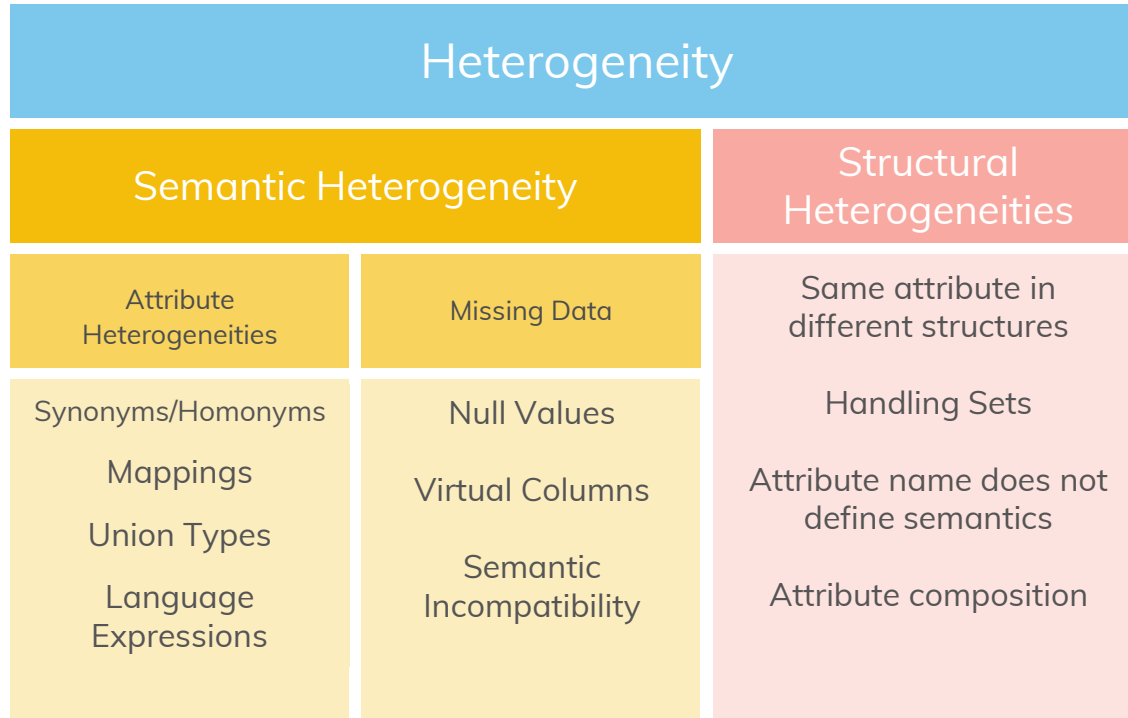
Virtual Columns

- Information is explicitly given in one schema but is derived in another schema.
(schema A: year of birth, age $\leftrightarrow$ schema B: year of birth, age = current year – year of birth)

Semantic Incompatibility

- A real-world concept is modelled in one schema, but not in the other.

Types of Heterogeneity



Same attribute in different structures

- The same/related attribute exists in different schemata at different positions (with different structure)
- A: *room* in *building* $\leftrightarrow$ B: *room* in *department* in *building*

Handling of sets

- A set of values can be expressed as one multi-value attribute in one schema and as a set of single-value attributes in another schema. (tutors $\leftrightarrow$ tutor1, tutor2,...)

Attribute name without semantics

- The attribute name does not describe its semantic adequately (Attribute11).

Composition of attributes

- The same information can be organized by one attribute or a set of attributes in the form of a hierarchy. („Lecture DIA Wed.“ $\leftrightarrow$ Lecture[title:DIA; day:Wed.])

Goals of Integration

Data consolidation / Homogenization

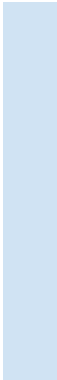
- Goal: global, consistent view on all systems
- Merge of data sets
- Data cleansing, error correction
- For the purpose of analysis / prediction / archiving ...

Interoperability

- Goal: data synchronization
- Interaction between independent systems and applications

Performance / Scalability / Availability

- Goal: Ensure quality of service
- HA (High Availability), Disaster Recovery
- Load Balancing (query performance vs. synchronization) and virtualization



Classification of Integration Approaches

Classification by Domain

GUI Integration

- Uniform visualisation / access to heterogeneous data sets
- E.g.: Portals, Mashups

Process Integration

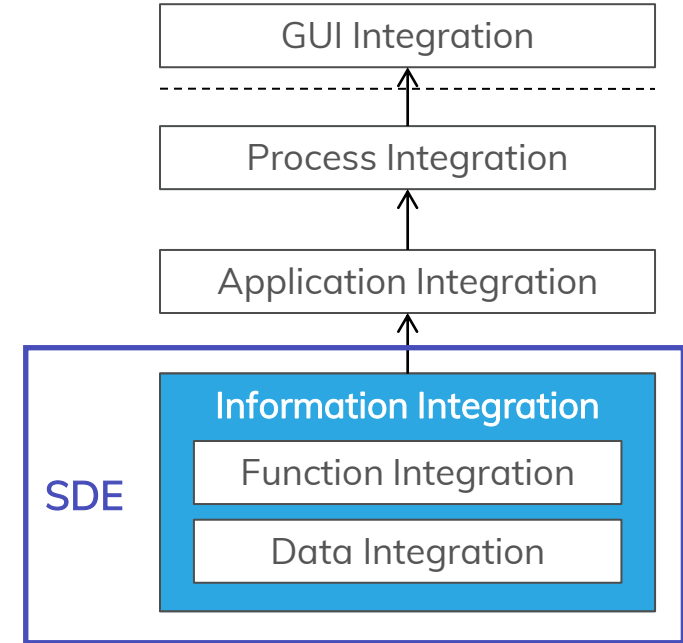
- Process components of homogeneous services
- E.g.: BPEL Engines, WSMS

Application Integration

- Integration of heterogeneous systems and applications
- E.g.: EAI Server, MOM, ETL Tools

Information Integration

- Queries on global schemata
- E.g.: VDBMS/FDBMS, ETL Tools, DSMS, PubSub, Replication



Classification by Time/Consistency

Synchronous (Strong Consistency)

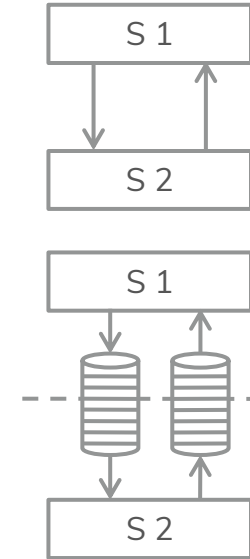
- Strong consistency between replicates
- Distributed transactions using 2PC (or 1PC, 3PC)
- Little local autonomy, data-driven integration

Asynchronous (Weak Consistency)

- Weak consistency allows asynchronous replication
- Batch-updates for efficient processing
- High performance and high local autonomy
- Latency between updates of primary copy and replicates
- Event model: ad hoc / data / time

Eventual Consistency (Special case of weak consistency)

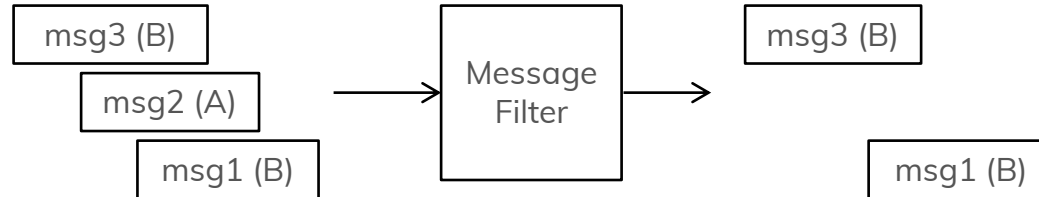
- Guarantees that system strives for consistent state, if no new updates occur
- Variants: Causal consistency, Read-your-writes, Session consistency, Monotonic reads/write



Data/Ad-Hoc (synchronous or asynchronous)

- Integration is triggered by updates of source systems or ad-hoc queries (distributed databases)
- Event – Condition – Action (ECA) principle
 - Event: Trigger, e.g. incoming query/message
 - Condition: evaluation of events, e.g. Priority=high
 - Action: operation or processing if condition is true

Example: Message
Filter:
E:'receive msg',
C:'type=A',
A:'delete'

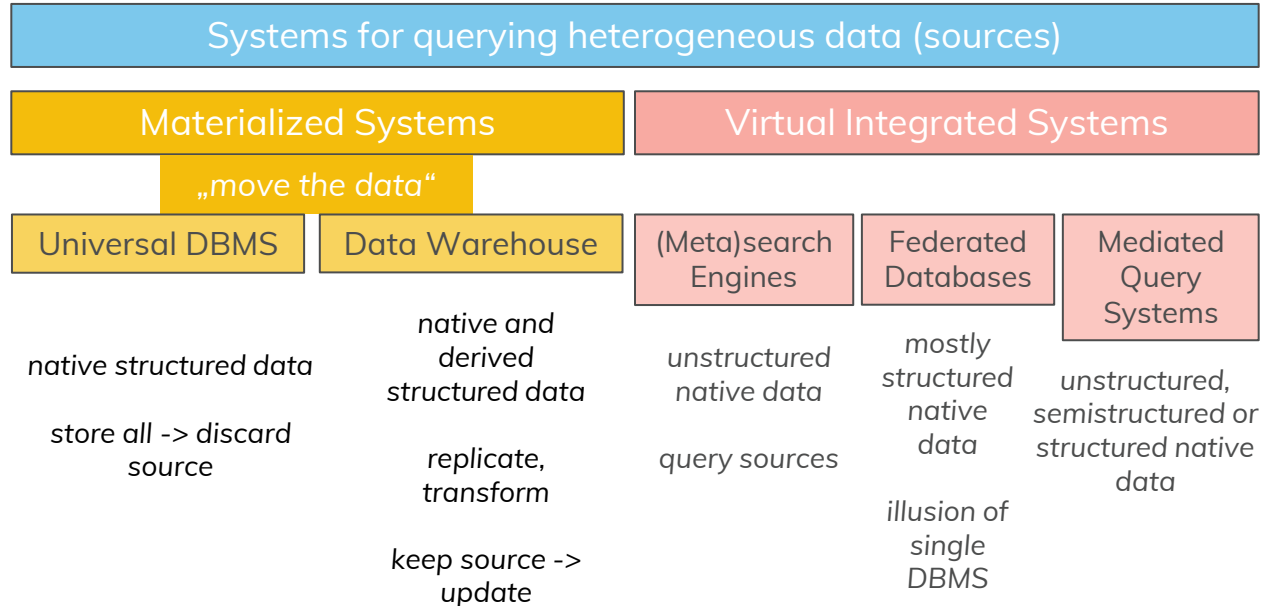


Time (asynchronous)

- Integration is triggered periodically
- Asynchronous processing with complete or incremental changes
- Typical for materialized integration (e.g. ETL)

Classification by Global Query Processing

Materialized vs. virtual integration



[Ruxandra Domenig, Klaus R. Dittrich: An Overview and Classification of Mediated Query Systems, SIGMOD Record, Vol. 28, No. 3, 1999]

Virtual vs. Materialized

Virtual (logic) integration

- Data is not available physically (only logical)
- Global queries must be translated into local queries for the source-systems

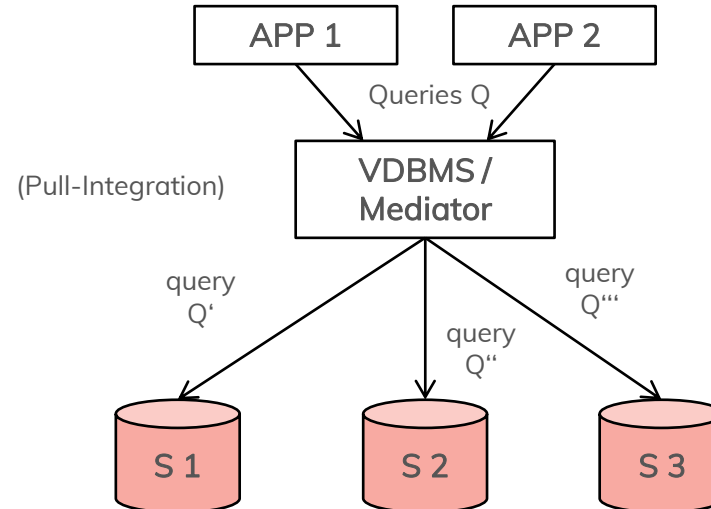
Pros

- Full flexibility
- Up-to-date
- No cost for synchronization

Cons

- 2PC for updates necessary
- Query performance
- No history/archiving

	VDBMS
data locality	virtual
time model	synchronous
event model	ad hoc / data
topology	hierarchy / hub



Distributed/Homogeneous databases – top down

- Top-Down design approach
 - One conceptual DB schema
 - (mostly) homogeneous system landscape (same vendor DBs, ...)
- DB-access like in a centralized case
- Limited / no autonomy of involved DBs
- Motivation: availability, performance, scalability

Federated/Heterogeneous databases– bottom up

- Consolidation of related information, distributed over several (heterogeneous) DBs
 - Designed and operated independently
- Characteristics
 - Heterogeneous data models and transaction management possible
 - Problems with heterogeneity/partial export of schema information
- Motivation: data consolidation and interoperability

Virtual vs. Materialized

Materialized (physical) integration

- Data is available physically
- Local queries on consolidated data

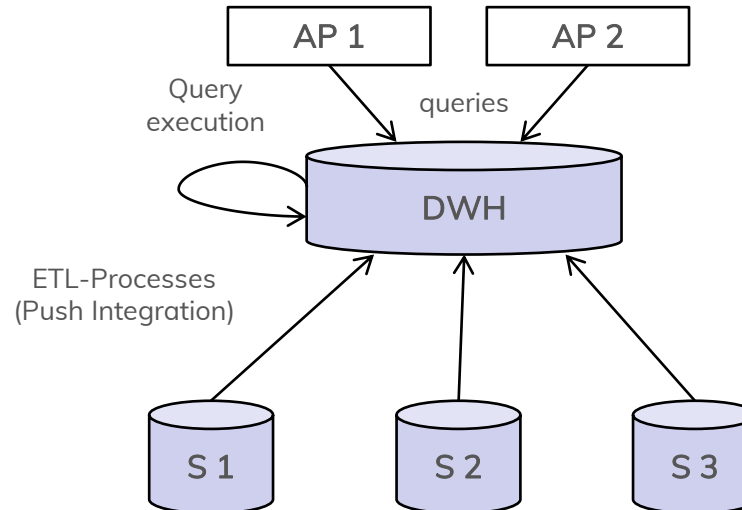
Pros

- Query performance (read-optimized)
- Independence from source systems
- history, archiving, consolidation

Cons

- Synchronization (up-to-date data)
- Flexibility/autonomy
- Data is kept redundantly

	ETL/DWH
data locality	materialized
time model	asynchronous
event model	data/time
topology	hierarchy / hub



The Remainder of the SDE Lecture

Topics we aim for...

- DWH Architecture
 - Multidimensional Modeling
 - SQL, OLAP and Multidimensional Queries
 - Indices & Views
 - Schema Integration
-
- Time Series Data
 - Graph Data
-
- Artificial Neural Networks
 - IR Models
 - Word Embeddings
-
- Approaching Analytics

